

**TRANSMITTAL LETTER TO THE  
UNITED STATES RECEIVING OFFICE**

Date	March 6, 2003
International Application	
Attorney Docket No.	P02,0120WO

**I. Certification under 37 CFR 1.10 (if applicable)**

EL 843731335US
Express Mail mailing number

Date of Deposit
-----------------

I hereby certify that the application/correspondence attached hereto is being deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to Assistant Commissioner for Patents, Washington, D.C. 20231.

*Craig G Thornton*  
Signature of person mailing correspondence

*Craig A Thornton*  
Typed or printed name of person mailing correspondence

**II. ☒ New International Application**

TITLE	INTERACTION DESIGN SYSTEM
-------	---------------------------

Earliest priority date (Day/Month/Year)
07 MARCH 2002

**SCREENING DISCLOSURE INFORMATION:** In order to assist in screening the accompanying international application for purposes of determining whether a license for foreign transmittal should and could be granted and for other purposes, the following information is supplied. (Note: check as many boxes as apply):

- A. ☐ The invention disclosed was not made in the United States.  
 B. ☐ There is no prior U.S. application relating to this invention.  
 C. ☒ The following prior U.S. application(s) contain subject matter which is related to the invention disclosed in the attached international application. (NOTE: priority to these applications may or may not be claimed on form PCT/RO/101 (Request) and this listing does not constitute a claim for priority.)

application no.	60/362,507	filed on	7 MARCH 2002
application no.		filed on	

- D. ☐ The present international application contains additional subject matter not found in the prior U.S. application(s) identified in paragraph C. above. The additional subject matter is found on pages \_\_\_\_\_ and ☐ DOES NOT ALTER ☐ MIGHT BE CONSIDERED TO ALTER the general nature of the invention in a manner which would require the U.S. application to have been made available for inspection by the appropriate defense agencies under 35 U.S.C. 181 and 37 CFR 5.1. See 37 CFR 5.15

**III. ☐ A Response to an Invitation from the RO/US. The following document(s) is(are) enclosed:**

- A. ☐ A Request for An Extension of Time to File a Response  
 B. ☐ A Power of Attorney (General or Regular)  
 C. ☐ Replacement pages:

pages		of the request (PCT/RO/101)	pages		of the figures
pages		of the description	pages		of the abstract
pages		of the claims			

- D. ☐ Submission of Priority Documents

Priority document		Priority document	
-------------------	--	-------------------	--

- E. ☐ Fees as specified on attached Fee Calculation sheet form PCT/RO/101 annex

**IV. ☐ A Request for Rectification under PCT 91 ☐ A Petition ☐ A Sequence Listing Diskette**

**V. ☐ Other (please specify):**

The person  
signing this  
form is the:

<input type="checkbox"/> Applicant	TREVOR JOIKE
<input checked="" type="checkbox"/> Attorney/Agent (Reg. No.)	Typed name of signer
<input type="checkbox"/> Common Representative	<i>Trevor Joike</i> Signature

## PCT REQUEST

P02,0120WO

Original (for SUBMISSION) - printed on 06.03.2003 11:08:55 AM

<b>0</b>	<b>For receiving Office use only</b>	
<b>0-1</b>	International Application No.	
<b>0-2</b>	International Filing Date	
<b>0-3</b>	Name of receiving Office and "PCT International Application"	
<b>0-4</b>	<b>Form - PCT/RO/101 PCT Request</b>	
<b>0-4-1</b>	Prepared using	<b>PCT-EASY Version 2.92 (updated 01.01.2003)</b>
<b>0-5</b>	<b>Petition</b> The undersigned requests that the present international application be processed according to the Patent Cooperation Treaty	
<b>0-6</b>	<b>Receiving Office (specified by the applicant)</b>	<b>United States Patent and Trademark Office (USPTO) (RO/US)</b>
<b>0-7</b>	<b>Applicant's or agent's file reference</b>	<b>P02,0120WO</b>
<b>I</b>	<b>Title of invention</b>	<b>INTERACTION DESIGN SYSTEM</b>
<b>II</b>	<b>Applicant</b>	
<b>II-1</b>	This person is:	<b>applicant only</b>
<b>II-2</b>	Applicant for	<b>all designated States except US</b>
<b>II-4</b>	Name	<b>HONEYWELL INC.</b>
<b>II-5</b>	Address:	<b>PO Box 2245 101 Columbia Road Morristown, NJ 07962-2245 United States of America</b>
<b>II-6</b>	State of nationality	<b>US</b>
<b>II-7</b>	State of residence	<b>US</b>
<b>II-8</b>	Telephone No.	<b>973-455-4259</b>
<b>II-9</b>	Facsimile No.	<b>973-455-2288</b>
<b>III-1</b>	<b>Applicant and/or inventor</b>	
<b>III-1-1</b>	This person is:	<b>applicant and inventor</b>
<b>III-1-2</b>	Applicant for	<b>US only</b>
<b>III-1-4</b>	Name (LAST, First)	<b>RAYMOND, Michelle, A</b>
<b>III-1-5</b>	Address:	<b>4131 Upton Avenue North Minneapolis, MN 55412 United States of America</b>
<b>III-1-6</b>	State of nationality	<b>US</b>
<b>III-1-7</b>	State of residence	<b>US</b>

## PCT REQUEST

P02,0120WO

Original (for SUBMISSION) - printed on 06.03.2003 11:08:55 AM

III-2	Applicant and/or inventor	
III-2-1	This person is:	applicant and inventor
III-2-2	Applicant for	US only
III-2-4	Name (LAST, First)	CARPENTER, Todd, P.
III-2-5	Address:	885 S. Lexington Parkway S St. Paul, MN 55116 United States of America
III-2-6	State of nationality	US
III-2-7	State of residence	US
III-3	Applicant and/or inventor	
III-3-1	This person is:	applicant and inventor
III-3-2	Applicant for	US only
III-3-4	Name (LAST, First)	MILLER, Christopher, A
III-3-5	Address:	1272 Raymond Avenue St. Paul, MN 55108 United States of America
III-3-6	State of nationality	US
III-3-7	State of residence	US
III-4	Applicant and/or inventor	
III-4-1	This person is:	applicant and inventor
III-4-2	Applicant for	US only
III-4-4	Name (LAST, First)	REISING, Dal Vernon, C
III-4-5	Address:	3393 Northdale Blvd. Apt. 234 Coon Rapids, MN 55448 United States of America
III-4-6	State of nationality	US
III-4-7	State of residence	US
III-5	Applicant and/or inventor	
III-5-1	This person is:	applicant and inventor
III-5-2	Applicant for	US only
III-5-4	Name (LAST, First)	KIFF, Liana-Maria
III-5-5	Address:	3771 Meadow Lane Excelsior, MN 55331 United States of America
III-5-6	State of nationality	US
III-5-7	State of residence	US

## PCT REQUEST

P02,0120WO

Original (for SUBMISSION) - printed on 06.03.2003 11:08:55 AM

IV-1	Agent or common representative; or address for correspondence The person identified below is hereby/has been appointed to act on behalf of the applicant(s) before the competent International Authorities as:	agent
IV-1-1	Name (LAST, First)	JOIKE, Trevor
IV-1-2	Address:	SCHIFF HARDIN & WAITE 233 S. Wacker Drive 6600 Sears Tower Chicago, IL 60606 United States of America
IV-1-3	Telephone No.	312-258-5774
IV-1-4	Facsimile No.	312-258-5921
IV-1-5	e-mail	tjoike@schiffhardin.com
IV-1-5	Agent's registration No.	25,542
V	Designation of States	
V-1	Regional Patent (other kinds of protection or treatment, if any, are specified between parentheses after the designation(s) concerned)	AP: GH GM KE LS MW MZ SD SL SZ TZ UG ZM ZW and any other State which is a Contracting State of the Harare Protocol and of the PCT EA: AM AZ BY KG KZ MD RU TJ TM and any other State which is a Contracting State of the Eurasian Patent Convention and of the PCT EP: AT BE BG CH&LI CY CZ DE DK EE ES FI FR GB GR HU IE IT LU MC NL PT SE SI SK TR and any other State which is a Contracting State of the European Patent Convention and of the PCT OA: BF BJ CF CG CI CM GA GN GQ GW ML MR NE SN TD TG and any other State which is a member State of OAPI and a Contracting State of the PCT
V-2	National Patent (other kinds of protection or treatment, if any, are specified between parentheses after the designation(s) concerned)	AE AG AL AM AT AU AZ BA BB BG BR BY BZ CA CH&LI CN CO CR CU CZ DE DK DM DZ EC EE ES FI GB GD GE GH GM HR HU ID IL IN IS JP KE KG KP KR KZ LC LK LR LS LT LU LV MA MD MG MK MN MW MX MZ NI NO NZ OM PH PL PT RO RU SC SD SE SG SK SL TJ TM TN TR TT TZ UA UG US UZ VC VN YU ZA ZM ZW

## PCT REQUEST

P02,0120WO

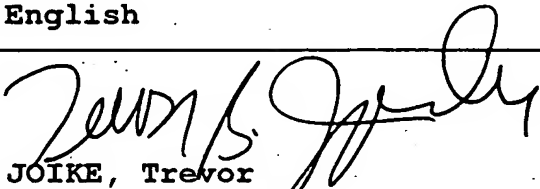
Original (for SUBMISSION) - printed on 06.03.2003 11:08:55 AM

<b>V-5</b>	<b>Precautionary Designation Statement</b> In addition to the designations made under items V-1, V-2 and V-3, the applicant also makes under Rule 4.9(b) all designations which would be permitted under the PCT except any designation(s) of the State(s) indicated under item V-6 below. The applicant declares that those additional designations are subject to confirmation and that any designation which is not confirmed before the expiration of 15 months from the priority date is to be regarded as withdrawn by the applicant at the expiration of that time limit.	
<b>V-6</b>	<b>Exclusion(s) from precautionary designations</b>	<b>NONE</b>
<b>VI-1</b>	<b>Priority claim of earlier national application</b>	
VI-1-1	Filing date	<b>07 March 2002 (07.03.2002)</b>
VI-1-2	Number	<b>60/362,507</b>
VI-1-3	Country	<b>US</b>
<b>VI-2</b>	<b>Priority document request</b> The receiving Office is requested to prepare and transmit to the International Bureau a certified copy of the earlier application(s) identified above as item(s):	<b>VI-1</b>
<b>VII-1</b>	<b>International Searching Authority Chosen</b>	<b>United States Patent and Trademark Office (USPTO) (ISA/US)</b>
<b>VIII</b>	<b>Declarations</b>	Number of declarations
VIII-1	Declaration as to the identity of the inventor	-
VIII-2	Declaration as to the applicant's entitlement, as at the international filing date, to apply for and be granted a patent	-
VIII-3	Declaration as to the applicant's entitlement, as at the international filing date, to claim the priority of the earlier application	-
VIII-4	Declaration of inventorship (only for the purposes of the designation of the United States of America)	-
VIII-5	Declaration as to non-prejudicial disclosures or exceptions to lack of novelty	-
<b>IX</b>	<b>Check list</b>	number of sheets
IX-1	Request (including declaration sheets)	<b>5</b>
IX-2	Description	<b>38</b>
IX-3	Claims	<b>14</b>
IX-4	Abstract	<b>1</b>
IX-5	Drawings	<b>4</b>
IX-7	<b>TOTAL</b>	<b>62</b>
		electronic file(s) attached
		<b>EZABST00.TXT</b>

## PCT REQUEST

P02,0120WO

Original (for SUBMISSION) - printed on 06.03.2003 11:08:55 AM

	Accompanying items	paper document(s) attached	electronic file(s) attached
IX-8	Fee calculation sheet	✓	-
IX-10	Original general power of attorney	✓	-
IX-17	PCT-EASY diskette	-	Diskette
IX-18	Other (specified):	Post Card Receipts and Transmittal Letter	-
IX-18	Other (specified):	Appendix A	-
IX-19	Figure of the drawings which should accompany the abstract	2	
IX-20	Language of filing of the international application	English	
X-1	Signature of applicant, agent or common representative		
X-1-1	Name (LAST, First)	JOIKE, Trevor	

## FOR RECEIVING OFFICE USE ONLY

10-1	Date of actual receipt of the purported international application	
10-2	Drawings:	
10-2-1	Received	
10-2-2	Not received	
10-3	Corrected date of actual receipt due to later but timely received papers or drawings completing the purported international application	
10-4	Date of timely receipt of the required corrections under PCT Article 11(2)	
10-5	International Searching Authority	ISA/US
10-6	Transmittal of search copy delayed until search fee is paid	

## FOR INTERNATIONAL BUREAU USE ONLY

11-1	Date of receipt of the record copy by the International Bureau	
------	---	--

**PCT (ANNEX - FEE CALCULATION SHEET)**

P02,0120WO

Original (for **SUBMISSION**) - printed on 06.03.2003 11:08:55 AM

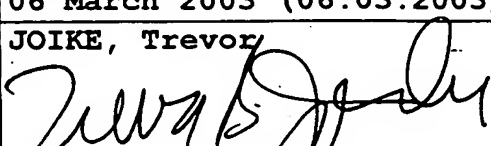
(This sheet is not part of and does not count as a sheet of the international application)

<b>0</b>	<b>For receiving Office use only</b>		
<b>0-1</b>	International Application No.		
<b>0-2</b>	Date stamp of the receiving Office		
<b>0-4</b>	<b>Form - PCT/RO/101 (Annex)</b>		
<b>0-4-1</b>	PCT Fee Calculation Sheet Prepared using	<b>PCT-EASY Version 2.92 (updated 01.01.2003)</b>	
<b>0-9</b>	Applicant's or agent's file reference	<b>P02,0120WO</b>	
<b>2</b>	Applicant	<b>HONEYWELL INC., et al.</b>	
<b>12</b>	<b>Calculation of prescribed fees</b>	fee amount/multiplier	Total amounts (USD)
<b>12-1</b>	Transmittal fee <b>T</b>	⇒	<b>240</b>
<b>12-2-1</b>	Search fee <b>S</b>	⇒	<b>700</b>
<b>12-2-2</b>	International search to be carried out by	<b>US</b>	
<b>12-3</b>	International fee		
	Basic fee (first 30 sheets) <b>b1</b>	<b>407</b>	
<b>12-4</b>	Remaining sheets	<b>32</b>	
<b>12-5</b>	Additional amount <b>(X)</b>	<b>9</b>	
<b>12-6</b>	Total additional amount <b>b2</b>	<b>288</b>	
<b>12-7</b>	<b>b1 + b2 = B</b>	<b>695</b>	
<b>12-8</b>	Designation fees		
	Number of designations contained in international application	<b>95</b>	
<b>12-9</b>	Number of designation fees payable (maximum 5)	<b>5</b>	
<b>12-10</b>	Amount of designation fee <b>(X)</b>	<b>88</b>	
<b>12-11</b>	Total designation fees <b>D</b>	<b>440</b>	
<b>12-12</b>	PCT-EASY fee reduction <b>R</b>	<b>-125</b>	
<b>12-13</b>	Total International fee (B+D-R) <b>I</b>	⇒	<b>1,010</b>
<b>12-14</b>	Fee for priority document		
	Number of priority documents requested	<b>1</b>	
<b>12-15</b>	Fee per document <b>(X)</b>	<b>20</b>	
<b>12-16</b>	Total priority document fee <b>P</b>	⇒	<b>20</b>
<b>12-17</b>	<b>TOTAL FEES PAYABLE (T+S+I+P)</b>	⇒	<b>1,970</b>
<b>12-19</b>	Mode of payment	<b>cheque</b>	
<b>12-20</b>	Deposit account instructions		
	The receiving Office:	<b>United States Patent and Trademark Office (USPTO) (RO/US)</b>	
<b>12-20-2</b>	Authorization to charge any deficiency or credit any overpayment in the total fees indicated above.	✓	
<b>12-20-3</b>	Authorization to charge the fee for priority document.	✓	

**PCT (ANNEX - FEE CALCULATION SHEET)**

P02,0120WO

Original (for **SUBMISSION**) - printed on 06.03.2003 11:08:55 AM

12-21	Deposit account No.	501519
12-22	Date	06 March 2003 (06.03.2003)
12-23	Name and signature	JOIKE, Trevor 

**VALIDATION LOG AND REMARKS**



## INTERACTION DESIGN SYSTEM

### Related Applications

This application claims the benefit of U.S. Provisional Application 60/362,507 filed March 7, 2002.

### 5 Technical Field of the Invention

The present invention relates to an interactive system that is useful in aiding a designer to design and generate user interfaces. For example, the interactive system is useful in aiding the designer to design and  
10 generate user interfaces for multiple devices (e.g., cellular telephones, internet browsers, personal digital assistants).

### Background of the Invention

15 Many tools have been created to aid a human in the design of products. For example, computer aided design (CAD), computer aided engineering (CAE), and computer aided manufacturing (CAM) systems have been developed to assist humans in the design and manufacture  
20 of various products. These tools, however, are not easily adaptable to a changing environment. That is, changing technologies relative to the products that can be designed by these tools soon make these tools obsolete. These tools also are not flexible. That is,

each of these tools is typically limited to a specific domain to which it applies and does not accommodate other domains. For example, many CAD systems that aid a designer in the design of integrated circuits do not also  
5 aid a designer in the design of automobile engines.

Moreover, these tools provide at most only a limited capability of designing an interface between the product or system being designed and the user of that product or system.

10           Accordingly, there is an interest in developing a design tool that can assist humans in the design of user interfaces in a more complex environment. Such a design tool should not be limited in the domains to which it applies or in the user interfaces that can be provided  
15 as an output of the design tool. Thus, the design tool, for example, should permit the design of user interfaces based on (i) the domains within which the user interface is to be used, (ii) the tasks that users will want to perform with respect to the user interface, (iii) the  
20 interaction delivery devices that can be used for the delivery of the user interface to the users, (iv) the roles, preferences, and limitations of the user or users who are to use the user interfaces, and/or (v) the presentation elements (i.e., display objects) to be used

to display input/output information to the user or users during use of the user interface being designed.

Existing design tools do not integrate these domain, task, available interaction delivery device, user, and presentation element considerations into a flexible interactive design tool that provides a comprehensive approach to information presentation and interaction, that enables the designer to quickly obtain and process information in a wide variety of environments and across a wide variety of tasks, that enables a designer to select from a variety of interaction delivery devices to accomplish tasks, that increases consistency and accuracy of the integration and dissemination of information by the use of common models, that accommodates changing equipment needs, and/or that provides user specific, device appropriate interactions. Moreover, existing design tools do not perform any reasoning with regard to the domain , task, interaction delivery device, user, and/or the presentation element considerations discussed above.

The present invention overcomes one or more of these or other problems.

#### Summary of the Invention

In accordance with one aspect of the present invention, a method of interactively designing a user interface comprises the following: receiving a domain model, a user model, a task model, and a device model, wherein the domain model characterizes an application for which the user interface is to be used, wherein the user model characterizes users who are to interface with the user interface, wherein the task model characterizes tasks to be performed between the user interface and the users, and wherein the device model characterizes interaction delivery devices that are available to deliver the user interface; and, matching characteristics in the domain model, the user model, the task model, and the device model so as to construct the user interface.

In accordance with another aspect of the present invention, a method of interactively designing a user interface comprises the following: creating a domain model, wherein the domain model contains information characterizing a designer selected application in a designer selected domain; creating a user model, wherein the user model contains information characterizing users of the user interface; creating a task model, wherein the task model contains task

primitives to be performed between the user and the user interface, and wherein the task model also contains types of information required by the task primitives; creating a device model, wherein the device model contains

5 information characterizing interaction delivery devices that are available to deliver the user interface; and, matching the information contained in the domain model, the user model, and the task model to the information contained in the device model and to presentation

10 elements contained in a presentation elements so as to construct the user interface, wherein the presentation elements comprise objects of the user interface that present information to the user.

In accordance with still another aspect of the

15 present invention, a method of interactively designing a user interface comprises the following: storing a domain model in a computer readable memory, wherein the domain model contains information characterizing data, concepts, and relations of an application in a domain as specified

20 by a designer; storing a user model in the computer readable memory, wherein the user model contains information characterizing roles and preferences of users of the user interface; storing a task model in the computer readable memory, wherein the task model contains

task primitives to be performed between the user and the user interface, information required of the task primitives, and sequences of the task primitives; storing a device model in the computer readable memory, wherein the device model contains information including modality characterizing interaction delivery devices that are available to deliver the user interface; matching the interaction delivery devices in the device model to the information required of the task primitives and to the information characterizing the users so as to identify interaction delivery devices that support the information requirements and the users; matching presentation elements to the task primitives and to the data, concepts, and relations of the domain model so as to identify ones of the presentation elements that support the task primitives and the data, concepts, and relations of the domain model; and, generating the user interface based on the identified interaction delivery device and the identified presentation elements.

20 In accordance with yet another aspect of the present invention, a method of interactively designing a system comprises the following: storing a domain model, a user model, a task model, and a device model in a computer readable memory, wherein the domain model

characterizes an application for which the system is to  
be used, wherein the user model characterizes a user who  
is to use the system, wherein the task model  
characterizes tasks to be performed between the system  
5 and the user, and wherein the device model characterizes  
devices to support the system; and, matching  
characteristics in the domain model, the user model, the  
task model, and the device model so as to construct the  
system.

10

Brief Description of the Drawings

These and other features and advantages will  
become more apparent from a detailed consideration of the  
invention when taken in conjunction with the drawings in  
15 which:

Figure 1 illustrates a computer useful in  
implementing an interaction design system according to an  
embodiment of the present invention;

20 Figure 2 illustrates the architecture of the  
interaction design system according to an embodiment of  
the present invention; and,

Figures 3A and 3B illustrate a flow chart of a  
program that can be used for the reasoning engine of  
Figure 2.

Detailed Description

Figure 1 illustrates a computer 10 that can be used to implement an interaction design system 12

5 according to one embodiment of the present invention. As shown in Figure 1, the computer 10 includes one or more input devices 14 such as a keyboard, a mouse, and/or a modem that can be used by a designer to provide the various inputs required by the interaction design system  
10 12 for the design and generation of user interfaces. For example, the designer can use a keyboard of the input devices 14 in providing these inputs to the interaction design system 12 residing on the computer 10.

Alternatively, one or more of these inputs can be  
15 generated remotely and supplied to the interaction design system 12 residing on the computer 10 through a modem of the input devices 14. As a further alternative, the designer can create the models and/or library discussed below on any suitable machine, save the models and/or  
20 library that the designer has created on a memory device, and load the contents of the memory device into the computer 10 at the appropriate time.

The computer 10 includes one or more output devices 16 such as a printer, a display, and a modem that



can be used by the designer in interacting with the interaction design system 12 executing on the computer 10. A common modem may be used as one of the input devices 14 and one of the output devices 16. The user  
5 interfaces designed and generated by the interaction design system 12 executing on the computer 10 can be provided to the designer as files in XML that the designer or others can then load on the interaction delivery devices selected by the interaction design  
10 system 12 to deliver (display visually, audibly, and/or otherwise) the user interfaces to the users.

The computer 10 also includes a memory 18 which may be in the form of random access memory, such as a floppy disk drive and/or a hard disk drive, and read only  
15 memory. The memory 18 stores the interaction design system 12 that is executed by the computer 10 to design and generate user interfaces, and may be used by the interaction design system 12 during its execution. The memory 18 may further be used to store the various inputs  
20 (models and library) that may be created by the designer and that are used by the interaction design system 12 during its execution to design and generate user interfaces..

Finally, the computer 10 includes a processor 20 that executes the interaction design system 12 stored by the memory 18.

Figure 2 illustrates the architecture of the interaction design system 12. The interaction design system 12 includes various inputs that the interaction design system 12 uses in designing and generating user interfaces. These inputs include a domain model 22, user models 24, task models 26, and device models 28. In addition, the interaction design system 12 includes a presentation elements library 30 that stores various presentation elements (objects) and a set of characteristics for each presentation element. The characteristics for each presentation element can be matched to the inputs created by the designer to allow a reasoning engine 32 of the interaction design system 12 to make qualitative judgments about the ability of the corresponding presentation elements of the presentation elements library 30 and the interaction delivery devices of the device models 28 to support the performance of the various interactions required of the user interfaces and the input and output of the information required by these interactions.

Thus, the reasoning engine 32 of the interaction design system 12 makes qualitative judgments about the ability of the presentation elements stored in the presentation element library 30 and the interaction delivery devices of the device models 28 (i) to support the application and domain specified by the designer in the domain model 22, (ii) to interact with the users as defined by the designer in the user models 24, (iii) to support the task primitives (i.e., the actions performed between the user interface and the users) as specified by the designer in the task models 26, and (iv) the information required by the task primitives as specified by the designer in the task models 26.

As indicated above, the designer creates the domain model 22 as an input of the interaction design system 12. More specifically, the designer creates the domain model 22 as a machine interpretable domain-specific representation of the relevant domain attributes to be given to a user interface. These attributes include the data, information, concepts, and/or relations pertinent to the application and domain for which the user interface is being designed. The form of the domain representation should be consistent with the other models that are created as inputs so that the interaction design

system 12 can properly match the characteristics and specifications of the interaction delivery devices of the device models 28 and of the presentation elements of the presentation elements library 30 with the characteristics  
5 and specifications provided in the domain model 22, the user models 24, and the task models 26. The domain model 22 should be application and domain specific.

In creating the domain model 22, the designer will already have in mind a particular application in a  
10 particular domain. For example, the designer may wish to design and generate user interfaces for the application of prescription drug ordering and filling in the domain of medicine. The designer must then model the data, information, concepts, and/or relations for this  
15 application. For example, in the application of prescription drug ordering and filling, the designer may determine that the user interfaces will deal with various items of information such as doctors, patients, pharmacists, medications, amounts, length of character  
20 strings required for the display of each of these entities, etc., with the relationships between these various entities.

The domain model 22 may be hierarchical having, for example, three levels in the hierarchy with domain

being the top level, domain elements being the next level down, and domain data being the bottom level. In the prescription drug ordering and filling example, the domain is prescription drug ordering and filling, the  
5 domain elements are doctors, patients, care givers, etc, and the domain data are amounts/values, labels/names, times such as pick-up times, lengths of character strings, etc.

In developing the domain model 22, the  
10 designer, for example, develops a meta-ontology that specifies the general required structure and vocabulary of the knowledge characterizing the designated domain, designs a specific schema for the selected application of the user interfaces being designed, and then populates  
15 the schema with the specific information, relationships, and concepts pertinent to this schema. The Resource Description Framework (RDF) notation may be used to create the schema and to populate it, although any other schema specific notation may be used for these purposes.  
20 Appendix A discloses an exemplary class hierarchy that may structure a domain-independent architecture for such a framework.

The RDF Specification is described in the documents "Resouce Description Framework (RDF) Model and Syntax

Specification: W3C Recommendation 22 February 1999", and  
"RDF/XML Syntax Specification (Revised):W3C Working Draft  
23 January 2003", both herein incorporated by reference.  
The domain model 22 created by the designer is stored in  
5 the memory 18 so that it is available to the interaction  
design system 12 during its execution on the computer 10.

The following is an abbreviated example of an  
RDF schema for the domain model 22 in the prescription  
drug ordering and filling application:

10

```
<rdfs:Class rdf:about = "Prescription"/>
<rdfs:Class rdf:about = "Medication"/>
<rdfs:Class rdf:about = "Pharmacy"/>
<rdf:Property rdf:about = "specifiesMedication"/>
15   <rdfs:domain rdf:resource = "Prescription"/>
      <rdfs:range rdf:about = "Medication"/>
<rdf:Property>
<rdf:Property rdf:about = "soldBy"/>
      <rdfs:domain rdf:resource = "Medication"/>
20   <rdfs:range rdf:resource = "Pharmacy"/>
<rdf:Property>
```

As can be seen, this RDF schema is only a partial schema  
for the prescription drug ordering and filling

application and should be expanded to include the other domain elements and domain data for a practical domain model.

In the user models 24 created by the designer, the designer captures the preferences, roles, and abilities (or limitations) of the users who are identified by the designer as the users of the user interfaces being designed. The preferences, roles, and abilities of the users can be captured using a flexible notation such as RDF. The preferences, roles, and abilities of the users include, for example, role descriptions, interaction delivery device preferences, modality (such as visual or audible) preferences, physical challenges that may confront the users of the user interface being designed, etc. Accordingly, the designer has the responsibility to understand the users and the application requirements so that the designer can create the user models 24 so that they define the relationships between the users and the application.

In the prescription drug ordering and filling example above, the users of the prescription drug ordering and filling interfaces may be any one or more of the following: doctors, patients, care givers, those who may be designated by the patients to pick up the

prescribed medicines, etc. The designer should keep in mind the preferences and abilities as well as the roles of all of these people in creating the user models 24. The user models 24 created by the designer are stored in the memory 18 so that they are available to the interaction design system 12 during its execution on the computer 10.

The following is an abbreviated example of an RDF schema for the user models 24 in the prescription drug ordering and filling application:

```
<rdfs:Class rdf:about="Person"/>
<rdfs:Class rdf:about="PhysicalAbilities"/>
<rdfs:Class rdf:about="VisualAcuity">
15   <rdfs:subClassOf rdf:resource="PhysicalAbilities"/>
</rdfs:Class>
<rdf:Property rdf:about="VisualRating">
    <rdfs:domain rdf:resource="VisualAcuity"/>
    <rdfs:range rdf:resource="Literal"/>
20   <allowedValues>0</allowedValues>
    <allowedValues>1</allowedValues>
    <allowedValues>2</allowedValues>
```



```
<rdf:Property rdf:about="vision">  
    <rdfs:domain rdf:resource="Person"/>  
    <rdfs:range rdf:resource="VisualAcuity"/>  
5 </rdf:Property>
```

As can be seen, this RDF schema is only a partial schema for the prescription drug ordering and filling application and should be expanded to include other  
10 preferences, roles, and abilities of the users who are to use the interfaces being designed.

In creating the task models 26, the designer captures the actions to be performed by the users when using the user interfaces being designed, the goals to be  
15 achieved by the user when using the user interfaces being designed, and the information required to perform the actions and achieve the goals. The task models 26 are meant to capture what actions on the part of the user the interfaces are intended to afford or support. The tasks  
20 can be captured using a flexible notation such as RDF that includes order-of-flow, nouns (i.e., the information required by the tasks), and the tasks that are to be performed. Accordingly, the designer has the responsibility to understand the task requirements within

the application and to apply the modeling notation to capture the specific task requirements.

In task modeling, the designer decomposes each task or interaction to be performed by the user into task primitives, an order to flow from each task primitive, and the type of information required by the task primitive. The task primitives may include any actions that a designer is likely to require between users and user interfaces for the relevant application in the relevant domain. For example, task primitives may include receive, instantiate, compare, monitor, assert, select, control, retract, change, detect, direct attention, navigate, adjust, etc. View, listen, review, and assess may be synonyms of the task primitive receive. Configure may be a synonym of instantiate. Observe and watch may be synonyms of the task primitive monitor. Set, enter, input, record, and declare may be synonyms of the task primitive assert. Pick and select-item-from-set may be synonyms of the task primitive select. Maintain, direct, and command may be synonyms of the task primitive control. Undo and withdraw may be synonyms of the task primitive retract. Modify, update, edit, and alter may be synonyms of the task primitive change. Identify, catch, and notice may be synonyms of the task primitive

detect. Focus may be a synonym of the task primitive direct attention. Move may be a synonym of the task primitive navigate. Configure may be a synonym of the task primitive adjust.

5           Also, as indicated above, the designer captures order-of-flow. Order-of-flow refers to the precedence between task primitives, and can be designated by junctions and links in the notation applied to the task models 26. Such junctions may include, for example, AND, 10 OR, and XOR junctions that may be synchronous or asynchronous. Links, for example, may be used to indicate that action B does not start before action A completes, that action A must be followed by action B, that action B must be preceded by action A, that actions 15 A and B are both required, etc.

Moreover, the task models 26 also include the information required by the task primitives of the tasks being modeled. For example, the task primitive receive requires that information be received by the user. The 20 designer defines the type of information for this task primitive in the task models 26. As another example, the task primitive instantiate requires that information be instantiated. The designer defines the type of information for this task primitive instantiation in the

task models 26. As still another example, the task primitive compare requires that certain information be compared to certain other information. The designer defines the type of information for this task primitive  
5 compare in the task models 26. The other task primitives also require information typing, and the designer defines the type of information for each of these task primitives that is used in the task models 26 as well. The task models 26 created by the designer are stored in the  
10 memory 18 so that they are is available to the interaction design system 12 during its execution on the computer 10.

In the prescription drug ordering and filling example above, the tasks to be performed may include  
15 directing the user to choose from among a displayed set of pharmacies, to choose whether the prescription is a refill, to designate a person to pick up the medication, to indicate that the prescription is to be moved from one pharmacy to another, etc.

20 The following is an abbreviated example of an RDF document adhering to the IDS RDF Schema for the task modeling 26 in the prescription drug ordering and filling application:

```
<userTask rdf:about="idsTask_00025"
    identifier="tskFillPrescription"
    name="Fill Prescription"
    primitive="NONPRIMITIVE"
5    rdfs:label="tskFillPrescription">
    <tasks rdf:resource="idsTask_00026"/>
    <entryTask rdf:resource="idsTask_00026"/>
    <tasks rdf:resource="idsTask_00027"/>
    <tasks rdf:resource="idsTask_00028"/>
10    <exitTask rdf:resource="idsTask_00029"/>
    <tasks rdf:resource="idsTask_00029"/>
    <tasks rdf:resource="idsTask_00032"/>
    <matchRequirements rdf:resource="idsTask_00036"/>
</userTask>
15 <junction rdf:about="idsTask_00026"
    identifier="AND1"
    operator="AND"
    primitive="JUNCTION"
    rdfs:label="AND1">
20    <parent rdf:resource="idsTask_00025"/>
    <mate rdf:resource="idsTask_00029"/>
    <followingRelations rdf:resource="idsTask_00030"/>
    <followingRelations rdf:resource="idsTask_00034"/>
</junction>
```

```
<userTask rdf:about="idsTask_00027"
    classElement="medication"
    identifier="tskSelectMedication"
    name="Select Medication"
5    primitive="SELECT"
    rdfs:label="tskSelectMedication">
    <parent rdf:resource="idsTask_00025"/>
    <precedingRelations rdf:resource="idsTask_00030"/>
    <followingRelations rdf:resource="idsTask_00031"/>
10    <matchRequirements rdf:resource="idsTask_00036"/>
</userTask>
...
<relation rdf:about="idsTask_00030"
    identifier="relation1"
15    relationship="PRECEDENCE"
    rdfs:label="relation1">
    <source rdf:resource="idsTask_00026"/>
    <destination rdf:resource="idsTask_00027"/>
</relation>
```

20

As can be seen, this RDF document is only a partial example for the prescription drug ordering and filling application and should be expanded to include other tasks

to be performed by the users when using the interfaces  
being designed.

In the device models 28, the designer captures  
the specifications and characteristics of the interaction  
5 delivery devices that are available to deliver the user  
interfaces being designed when these user interfaces are  
invoked by the applications for which they are designed.  
These specifications should include the capabilities and  
modalities that are supported by the available  
10 interaction delivery devices and that are relevant to the  
application. The capabilities, for example, may include  
bandwidth, memory, screen, lines of display, width of  
display, illumination, etc., and the modalities, for  
example, may include visual, audible, etc.

15 These specifications and characteristics of the  
interaction delivery devices can be captured using a  
flexible notation such as RDF that includes a mechanism  
to describe an interaction delivery device's specific  
input and output modalities and capabilities. The  
20 designer has the responsibility to understand the  
interaction delivery device specification and to apply  
the interaction delivery device description notation to  
produce the device models 28. The device models 28  
created by the designer are stored in the memory 18 so

that they are available to the interaction design system  
12 during its execution on the computer 10.

In the prescription drug ordering and filling  
example above, interaction delivery devices can include  
5 web browsers, personal digital assistants, telephonic  
interaction delivery devices, etc. These interaction  
delivery devices are existing interaction delivery  
devices that can be used to deliver the user interface to  
the users. In the case of audio interaction delivery  
10 devices, the capabilities of such interaction delivery  
devices can include number of tones, word rate, speech  
vs. tone only, etc. In the case of visual interaction  
delivery devices, the capabilities of such interaction  
delivery devices can include number of lines, number of  
15 characters per line, update rate, etc. In the case of  
hardware buttons, the capabilities can include  
yes/no/select vs. none, numeric vs. none, up/down vs.  
none, left/right vs. none, alphabetic vs. none, input  
rate, etc.

20 The following is an abbreviated example of an  
RDF schema for the device models 28 in the prescription  
drug ordering and filling application:

```
<rdfs:Class rdf:about="InteractionDevice">
```



```

    <rdfs:subClassOf rdf:resource="DomainItem"/>
</rdfs:Class>

<rdfs:Class rdf:about="DeviceSignature">
    <rdfs:subClassOf rdf:resource="NounSignature">
5  </rdfs:Class>

<rdfs:Class rdf:about="AudioDisplayCharacteristics"/>
<rdfs:Class rdf:about="HardwareButtonCharacteristics"/>
<rdf:Property rdf:about="supportedDeviceSignature">
    <rdfs:range rdf:resource="DeviceSignature"/>
10 <rdfs:domain rdf:resource="InteractionDevice"/>
</rdf:Property>

<rdf:Property rdf:about="audioDisplay">
    <rdfs:range
rdf:resource="AudioDisplayCharacteristics"/>
15 <rdfs:domain rdf:resource="DeviceSignature"/>
</rdf:Property>

<rdf:Property rdf:about="speechOutput"
    <rdfs:range="boolean">
    <rdfs:domain
20 rdf:resource="AudioDisplayCharacteristics"/>
    <rdfs:range rdf:resource="&rdfs;Literal"/>
</rdf:Property>
```

As can be seen, this RDF schema is only a partial schema for the description of interaction devices and should be expanded to include other interaction delivery device specifications for the interaction delivery devices that  
5 can potentially deliver user interfaces to users.

Presentation elements are the display objects that are used to present information to or acquire information from a user of the user interface being designed. In the context of a web browser being used as  
10 the interaction delivery device, a presentation element may be a pop up menu, a pull down menu, a dialog box, a button, etc.

Each of the presentation elements stored in the presentation elements library 30 contains a set of  
15 functionality and usability characteristics that the corresponding presentation element either supports or requires for correct application in a presentation. The functionality and usability characteristics describe the quality of the interaction that can be achieved given the  
20 functionality and usability characteristics of the display objects. The functionality and usability characteristics for each presentation element should have a form so that they can be matched by the reasoning engine 32 to the other inputs, and so that the reasoning

engine 32 can make qualitative judgments about the ability of the presentation elements to support the input and output of the data to be exchanged between the users and the user interfaces.

5           In the prescription drug ordering and filling application, examples of presentation elements include pull down menus, dialog boxes, windows, buttons, etc.

          The following is an abbreviated example of an XML composer, written in Java, for the presentation  
10 elements library 30 in the prescription drug ordering and filling application:

```
private String createXML(final Presentation
presentation) {
15       // get the task's name

      String taskName = "name=" + "\"" +
presentation.getInteractionRequirement().getTask().getNam
e() + "\"" ;

      // get the task's identifier
20       String taskIdentifier = "identifier=" + "\"" +
presentation.getInteractionRequirement().getTask().getId
entifier() + "\"" ;

      // get the domain element content
```

```
        output =  
presentation.getInteractionRequirement().getDomainItem().  
getContent(presentation.getInteractionRequirement().getTa  
sk());  
5      // Create outer <textbox> element tags  
      String preFix = "<textBox " + taskName + " " +  
taskIdentifier + ">";  
      preFix = preFix + " <label><text>" +  
presentation.getInteractionRequirement().getTask().getNam  
10 e() + "</text></label>";  
      //sandwich content between open and end tags  
      output = preFix + output + "</textBox>";  
      return output;  
    }
```

15

As can be seen, this XML composer is only one example and should be expanded to include other presentation elements that can potentially be used to exchange information between the users and the application for which the user  
20 interfaces are being designed.

As shown in Figure 2, the domain model 22, the user models 24, the task models 26, and the device models 28 combine to define the interaction requirements of the user interface being designed. Each interaction

requirement is a combination of a task primitive and information required by the task primitive as influenced by the characteristics of the users and the application and domain for which the user interface is being  
5 designed. Thus, a user interface typically involves a plurality of interaction requirements that define the totality of the way in which the users interact with the user interface being defined.

The domain model 22, the user models 24, the  
10 task models 26, the device models 28, and the presentation elements library 30 are stored in the memory 18 in preparation for the execution of the reasoning engine 32. Based on the domain model 22, the user models 24, the task models 26, the device models 28, and the  
15 presentation elements library 30, the reasoning engine 32 makes qualitative judgments about the best fit between the presentation elements and the interaction delivery devices in order to interact with the user through the user interface as intended by the designer.

20 A flow chart of a program that can be used for the reasoning engine 32 is shown in Figures 3A and 3B. Once the domain model 22, the user models 24, the task models 26, the device models 28 have been created and entered by the designer and stored in the memory 18, and

once the presentation elements library 30 has been populated with the presentation elements, the reasoning engine 32 may be executed.

Accordingly, a block 40 filters the interaction  
5 delivery devices of the device models 28 based on the information requirements of the task primitives defined in the tasks models 26 and the roles, preferences, and abilities of the users as defined in the user models 24 so as to form an intersection between these available  
10 interaction delivery devices, information requirements, and user characteristics. Accordingly, the information requirements of the task primitives as modeled by the designer in the tasks models 26, the roles, preferences, and abilities of the users as modeled by the designer in  
15 the user models 24, and the characteristic specifications and capabilities of the interaction delivery devices as modeled by the designer in the device models 28 are matched, and only those available interaction delivery devices that can support those information requirements  
20 and user characteristics are saved for further processing. These saved interaction delivery devices, therefore, are the filtered output of the filtering process of the block 40.

As an example, if a user is to invoke an interface to change the temperature of a room, the block 40 of the interaction design system 12 should consider those interaction delivery devices that are available to  
5 change temperatures and that meet the preferences, roles and abilities of the intended users.

A block 42 filters the presentation elements stored in the presentation library 30 based on the task primitives of the task models 26 and the characteristics  
10 of the information, concepts, and relations of the domain model 22 to form an intersection between the presentation elements stored in the presentation library 30, the task primitives of the task models 26, and the domain characteristics of the domain model 22. Accordingly, the  
15 presentation elements, the task primitives, and the domain characteristics are matched, and only those presentation elements that can support the task primitives and domain characteristics are saved for further processing. These saved presentation elements,  
20 therefore, are the output of the filtering process of the block 42.

As an example, if a task is a SELECT primitive, and the user is supposed to select a numeric value as a domain characteristic, the block 42 considers those

presentation elements that support SELECT of numeric lists.

A block 44 creates a presentation comprising each presentation element saved as a result of the  
5 processing of the block 42 and the interaction delivery devices that are saved as a result of the processing of the block 40 and that support the corresponding presentation element. Accordingly, the block 44 matches the interaction delivery devices saved by the block 40  
10 and the presentation elements saved by the block 42, and creates a presentation for each saved presentation element and matching interaction delivery device.

It should be understood that each presentation element is not required to support all of the interaction  
15 delivery devices. Similarly, each interaction delivery device is not required to support all presentation elements. However, it is quite possible that any given presentation element saved by the block 42 will match more than one of the interaction delivery devices saved  
20 by the block 40. Therefore, a presentation element may be included in more than one presentation. Each presentation comprises a presentation element and an interaction delivery device as a matching pair. The



presentations at the output of the block 44 form a fully-resolved set of usability or interaction capabilities.

A block 46 assigns a score to each presentation based on the quality of the match that produced the  
5 interaction delivery devices saved by the block 40 and the quality of the match that produced the presentation elements saved by the block 42. This scoring is a qualitative judgment about how well a presentation (presentation element/interactive deliver device pair)  
10 meets the domain characteristics of the domain model 22, the roles, preferences, and abilities of the users as defined in the user models 24, and the task primitives and corresponding information requirements defined in the task models 26.

15 For example, if a presentation element is to perform the task primitive receive in order to display information such as current temperature to a user, various presentation elements, such as a digital readout, a round graphic, and a thermostat graphic, can be  
20 considered. Each of these presentation elements may be assigned a set of values such as DisplayScope (DS), DisplayResolution (DR), DisplayBandwidth, DisplayBandwidth (DB), DisplayImportance (DI), DisplayObtrusiveness (DO), ControlScope (CS),

ControlResolution (CR), ControlBandwidth (CB), and/or  
ControlImportance (CI) that describe the characteristics  
of the presentation element. Similarly, the information  
requirement of this presentation element may also be  
5 assigned a set of values such as DisplayScope (DS),  
DisplayResolution (DR), DisplayBandwidth,  
DisplayBandwidth (DB), DisplayImportance (DI),  
DisplayObtrusiveness (DO), ControlScope (CS),  
ControlResolution (CR), ControlBandwidth (CB), and/or  
10 ControlImportance (CI) that describe the characteristics  
of the presentation element. Some presentation elements  
are essentially only displays, showing information,  
(hence the use of the word "display"). Other  
presentation elements allow for inputs or manipulations  
15 (hence the use of the word "control"). Still other  
presentation elements might be both a display and a  
control. In any case, a set of values DS-CI need to be  
assigned to the presentation elements so that the  
reasoner of Figures 3A and 3B can reason. These values  
20 may be compared, and a score can be generated that  
indicates how closely the values of the information  
requirement and the presentation element match. The  
block 46 uses these scores for the corresponding  
presentations. It is also possible to generate a similar

score for each interaction delivery device that indicates how closely the characteristics of the interaction delivery devices match the characteristics stored in the user models 24 and the tasks models 26. The block 46 may  
5 be arranged to combine the presentation element score and the interaction delivery device score to form a composite score for the corresponding presentation.

A block 48 then sorts the presentations by score. For each interaction requirement of the user  
10 interface being designed as defined by the domain model 22, the user models 24, the task models 26, and the device models 28, a block 50 selects the presentations having the best scores. For each such interaction requirement, a block 52 chooses a presentation off of the  
15 best score list. If the designer is not satisfied with the collection of presentations as indicated by a block 54, the designer may change the characteristics and/or requirements of one or more of the models at a block 56 until the designer is satisfied with the resulting user  
20 interface. Alternatively or additionally, the designer may cause the interaction design system 12 to choose a different combination of presentations at the block 52. Accordingly, the design of a user interface is an iterative process between the designer and the

interaction design system 12, and this process continues until the designer is satisfied with the designed user interface.

A block 58 generates the presentations as an XML file, and a block 60 applies the interaction delivery devices' XSL to the XML generated by the block 58. It is noted that the result of the block 60 does not include the necessary application-specific communications between the chosen interaction delivery device and the application for which the user interface is being designed. Rather, the result of the block 60 is the interaction appropriate for the interaction delivery device once an appropriate application makes the user interface available to the user.

The device models 28 and the presentation elements library 30 may remain unchanged from user interface design to user interface design. On the other hand, the designer may change the device models 28 and/or the presentation elements library 30 in preparation for the design of one or more user interfaces. The domain model 22, the user models 24, and/or the task models 26 will likely, although not necessarily, change from design to design. One of the advantages of the interaction design system 12 is that it can adapt to new domains, to

new applications, to new tasks, to new interaction  
delivery devices, to new presentation elements, and to  
new user requirements simply by storing the appropriate  
information in the domain model 22, the user models 24,  
5 the task models 26, the device models 28, and/or the  
presentation elements library 30.

Certain modifications of the present invention  
have been described above. Other modifications will  
occur to those practicing in the art of the present  
10 invention. For example, although the present invention  
is specifically described above in terms of designing and  
generating user interfaces, the present invention may be  
adapted to aiding a designer in the design and generation  
of products, systems, machines, and arrangements other  
15 than user interfaces.

Also, the flow chart of Figures 3A and 3B is  
described above (particularly with respect to blocks 40-  
56) as generating all presentations for all interaction  
requirements prior to generation of the XML file.

20 Instead, a presentation can be added to the XML file one  
interaction requirement at a time until all interaction  
requirements are designed into the user interface

Accordingly, the description of the present  
invention is to be construed as illustrative only and is

for the purpose of teaching those skilled in the art the  
best mode of carrying out the invention. The details may  
be varied substantially without departing from the spirit  
of the invention, and the exclusive use of all  
5 modifications which are within the scope of the appended  
claims is reserved.

WE CLAIM:

1           1.    A method of interactively designing a user  
2 interface comprising:  
3           receiving a domain model, a user model, a task  
4 model, and a device model, wherein the domain model  
5 characterizes an application for which the user interface  
6 is to be used, wherein the user model characterizes users  
7 who are to interface with the user interface, wherein the  
8 task model characterizes tasks to be performed between  
9 the user interface and the users, and wherein the device  
10 model characterizes interaction delivery devices that are  
11 available to deliver the user interface; and,  
12           matching characteristics in the domain model,  
13 the user model, the task model, and the device model so  
14 as to construct the user interface.

1           2.    The method of claim 1 wherein the matching  
2 of characteristics comprises forming an intersection  
3 between the domain model, the user model, the task model,  
4 and the device model.

1                   3.    The method of claim 1 wherein the matching  
2   of characteristics comprises:

3                   matching the interaction delivery devices to  
4   information requirements defined in the tasks model and  
5   to the users defined in the user models to identify  
6   interaction delivery devices that support the information  
7   requirements and the users; and,

8                   matching presentation elements to task  
9   primitives of the task model and to characteristics  
10   provided in the domain model to identify presentation  
11   elements that support the task primitives and the domain  
12   characteristics, wherein the presentation elements  
13   comprise display objects.

1                   4.    The method of claim 3 wherein the matching  
2   of characteristics comprises creating a presentation for  
3   each identified presentation element and a matching one  
4   of the identified interaction delivery devices.

1                   5.    The method of claim 4 wherein the matching  
2   of characteristics comprises scoring and sorting the  
3   presentations, and wherein the matching of  
4   characteristics comprises selecting the presentations  
5   having the best scores.



1           6.    The method of claim 5 wherein the matching  
2 of characteristics comprises generating the user  
3 interface based on the selected presentations.

1           7.    The method of claim 5 wherein the  
2 selecting of the presentations comprises selecting the  
3 presentations having the best scores for all interactions  
4 between the users and the user interface.

1           8.    The method of claim 7 wherein the matching  
2 of characteristics comprises generating the user  
3 interface based on the selected presentations.

1           9.    The method of claim 4 wherein the matching  
2 of characteristics comprises generating the user  
3 interface based on the presentations.

1           10.   A method of interactively designing a user  
2 interface comprising:  
3                creating a domain model, wherein the domain  
4 model contains information characterizing a designer  
5 selected application in a designed selected domain;

6           creating a user model, wherein the user model  
7 contains information characterizing users of the user  
8 interface;

9           creating a task model, wherein the task model  
10 contains task primitives to be performed between the user  
11 and the user interface, and wherein the task model also  
12 contains types of information required by the task  
13 primitives;

14           creating a device model, wherein the device  
15 model contains information characterizing interaction  
16 delivery devices that are available to deliver the user  
17 interface; and,

18           matching the information contained in the  
19 domain model, the user model, and the task model to the  
20 information contained in the device model and to  
21 presentation elements contained in a presentation  
22 elements so as to construct the user interface, wherein  
23 the presentation elements comprise objects of the user  
24 interface that present information to the user.

1           11. The method of claim 10 wherein the domain  
2 model, the user model, the task model, and the device  
3 model are created using a consistent notation.

1           12. The method of claim 11 wherein the  
2 notation adheres to the Resource Description Framework  
3 specification or other specific knowledge technology  
4 notations.

1           13. The method of claim 10 wherein the domain  
2 model, the user model, the task model, and the device  
3 model are stored in a computer readable memory.

1           14. The method of claim 10 wherein the  
2 matching of the information comprises forming an  
3 intersection between the presentation elements and the  
4 information contained in the domain model, the user  
5 model, the task model, the device model, and the  
6 presentation elements library.

1           15. The method of claim 10 wherein the  
2 matching of the information comprises:  
3           matching the interaction delivery devices to  
4 the type of information required of the task primitives  
5 and to the information characterizing the users so as to  
6 identify interaction delivery devices that support the  
7 information requirements and the users; and,  
8           matching the presentation elements to the task  
9 primitives and to the information characterizing the  
10 designer selected application in the designer selected  
11 domain so as to identify presentation elements that  
12 support the task primitives and the domain information.

1           16. The method of claim 15 wherein the  
2 matching of the information comprises creating a  
3 presentation for each identified presentation element  
4 that matches at least one of the identified interaction  
5 delivery devices.

1           17. The method of claim 16 wherein the  
2 matching of the information comprises scoring and sorting  
3 the presentations, and wherein the matching of the  
4 information comprises selecting the presentations having  
5 the best scores.

1           18. The method of claim 17 wherein the  
2 matching of the information comprises generating the user  
3 interface based on the selected presentations.

1           19. The method of claim 17 wherein the  
2 selecting of the presentations comprises selecting the  
3 presentations having the best scores for all interactions  
4 to be performed by the user interface.

1           20. The method of claim 19 wherein the  
2 matching of the information comprises generating the user  
3 interface based on the selected presentations.

1           21. The method of claim 10 wherein the domain  
2 model, the user model, the task model, and the device  
3 model are created using a consistent notation, and  
4 wherein the matching of the information comprises:

5           matching the interaction delivery devices to  
6 the information required of the task primitives and to  
7 the information characterizing the users so as to  
8 identify interaction delivery devices that support the  
9 information requirements and the users; and,

10           matching the presentation elements to the task  
11 primitives and to the information characterizing a  
12 specific application in a specific domain so as to  
13 identify presentation elements that support the task  
14 primitives and the domain information.

1           22. The method of claim 21 wherein the  
2 matching of the information comprises creating  
3 presentations, and wherein each presentation comprises a  
4 matching pair of one of the presentation elements and one  
5 of the interaction delivery devices.

1           23. The method of claim 22 wherein the  
2 matching of characteristics comprises selecting one of

3 the presentations for each interaction to be performed  
4 between the user interface and the users.

1 24. A method of interactively designing a user  
2 interface comprising:

3 storing a domain model in a computer readable  
4 memory, wherein the domain model contains information

5 characterizing data, concepts, and relations of an  
6 application in a domain as specified by a designer;

7 storing a user model in the computer readable  
8 memory, wherein the user model contains information  
9 characterizing roles and preferences of users of the user  
10 interface;

11 storing a task model in the computer readable  
12 memory, wherein the task model contains task primitives  
13 to be performed between the user and the user interface,  
14 type of information required of the task primitives, and  
15 sequences of the task primitives;

16 storing a device model in the computer readable  
17 memory, wherein the device model contains information  
18 including modality characterizing interaction delivery  
19 devices that are available to deliver the user interface;

20           matching the interaction delivery devices in  
21 the device model to the type of information required of  
22 the task primitives and to the information characterizing  
23 the users so as to identify interaction delivery devices  
24 that support the information requirements and the users;  
25           matching presentation elements to the task  
26 primitives and to the data, concepts, and relations of  
27 the domain model so as to identify ones of the  
28 presentation elements that support the task primitives  
29 and the data, concepts, and relations of the domain  
30 model; and,  
31           generating the user interface based on the  
32 identified interaction delivery device and the identified  
33 presentation elements.

1           25. The method of claim 24 wherein the  
2 generating of the user interface comprises creating  
3 presentations between matching ones of the identified  
4 presentation elements and ones of the identified  
5 interaction delivery devices.

1           26. The method of claim 25 wherein the  
2 generating of the user interface comprises scoring and  
3 sorting the presentations, and wherein the generating of



4 the user interface comprises selecting the presentations  
5 having the best scores.

1           27. The method of claim 26 wherein the  
2 generating of the user interface comprises generating the  
3 user interface based on the selected presentations.

1           28. The method of claim 26 wherein the  
2 selecting of the presentations comprises selecting the  
3 presentations having the best scores for all interactions  
4 to be performed by the user interface.

1           29. The method of claim 28 wherein the  
2 generating of the user interface comprises generating the  
3 user interface based on the selected presentations.

1           30. The method of claim 29 further comprising  
2 creating the domain model, the user model, the task  
3 model, and the device model using a consistent notation.

1           31. The method of claim 30 wherein the  
2 notation adheres to the Resource Description Framework  
3 specification or other specific knowledge technology  
4 notations.

1           32. The method of claim 30 wherein the  
2 generating of the user interface comprises creating  
3 presentations between matching ones of the identified  
4 presentation elements and the identified interaction  
5 delivery devices.

1           33. The method of claim 32 wherein the  
2 generating of the user interface comprises selecting one  
3 of the presentations for each interaction to be performed  
4 between the user interface and the users.

1           34. The method of claim 33 wherein the  
2 generating of the user interface comprises generating the  
3 user interface based on the selected presentations.

1           35. The method of claim 24 wherein the  
2 generating of the user interface comprises creating  
3 presentations between matching ones of the identified  
4 presentation elements and the identified interaction  
5 delivery devices.

1           36. The method of claim 35 wherein the  
2 generating of the user interface comprises selecting one

3 of the presentations for each interaction to be performed  
4 between the user interface and the users.

1           37. The method of claim 36 wherein the  
2 generating of the user interface comprises generating the  
3 user interface based on the selected presentations.

1           38. A method of interactively designing a  
2 system comprising:  
3           storing a domain model, a user model, a task  
4 model, and a device model in a computer readable memory,  
5 wherein the domain model characterizes an application for  
6 which the system is to be used, wherein the user model  
7 characterizes a user who is to use the system, wherein  
8 the task model characterizes tasks to be performed  
9 between the system and the user, and wherein the device  
10 model characterizes devices to support the system; and,  
11           matching characteristics in the domain model,  
12 the user model, the task model, and the device model so  
13 as to construct the system.

1           39. The method of claim 38 wherein the  
2 matching of characteristics comprises forming an

3 intersection between the domain model, the user model,  
4 the task model, and the device model.

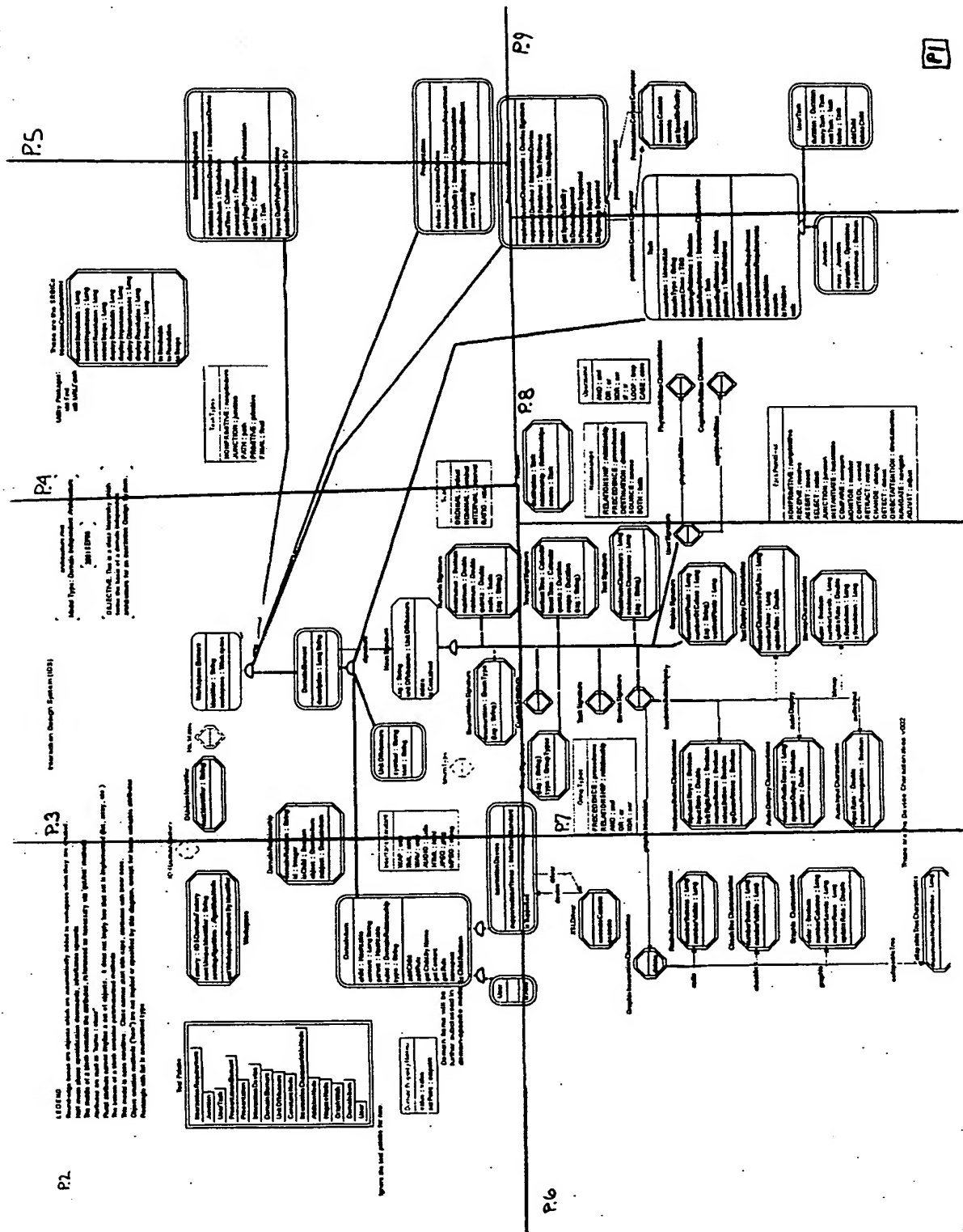
1           40. The method of claim 39 further comprising  
2 creating the domain model, the user model, the task  
3 model, and the device model using a consistent notation.

1           41. The method of claim 40 wherein the  
2 notation adheres to the Resource Description Framework  
3 specification or other specific knowledge technology  
4 notations.

Abstract

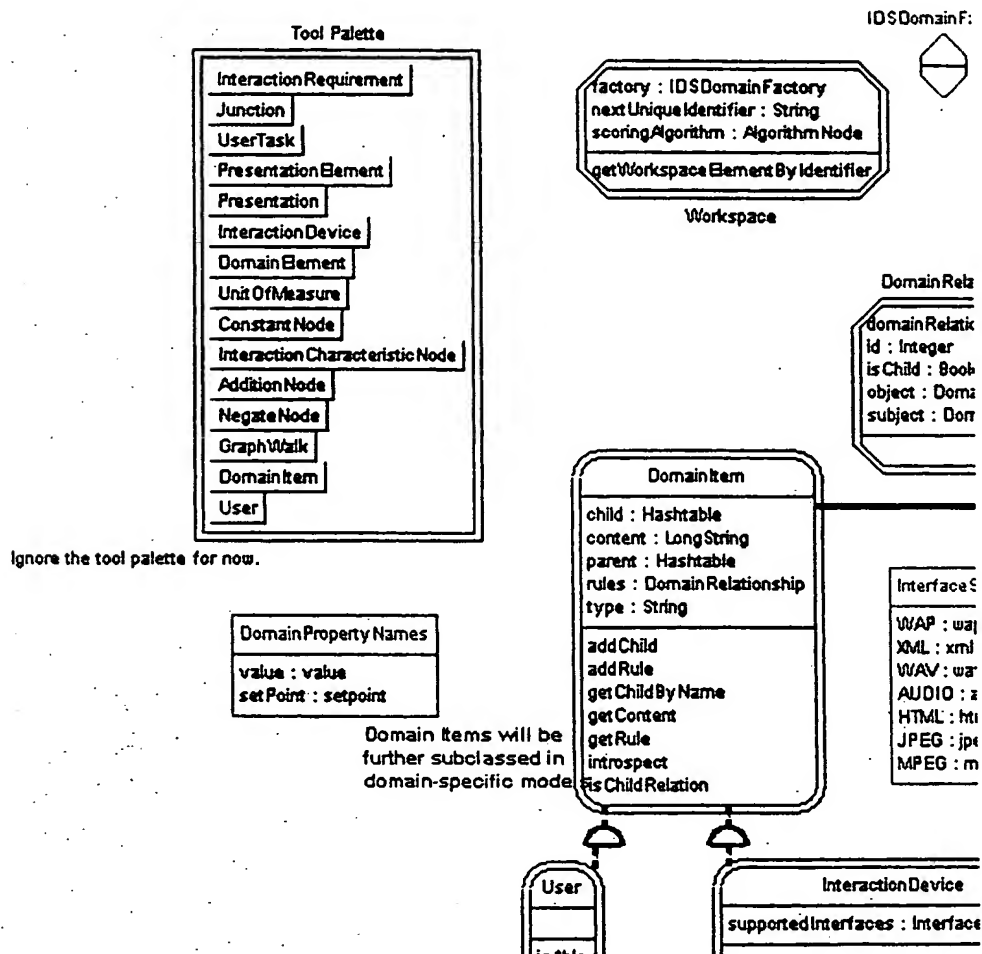
1           An interaction design system may be used by a  
2 designer to design a user interface. The designer  
3 supplies the interaction design system with a domain  
4 model that contains information characterizing an  
5 application in a domain, a user model that contains  
6 information characterizing the users of the user  
7 interface, a task model that contains task primitives to  
8 be performed between the user and the user interface and  
9 the type of information required by the task primitives,  
10 and a device model that contains information  
11 characterizing the interaction delivery devices that are  
12 available to deliver the user interface. The interaction  
13 design system then matches the interaction delivery  
14 devices in the device model to the type of information  
15 required by the task primitives and to the information  
16 characterizing the users, matches presentation objects to  
17 the task primitives and to the information of the domain  
18 model, and generates the user interface based on the  
19 matches.

Appendix A  
Interaction Design System Architecture



## LEGEND

Round-edge boxes are objects which are automatically added to workspace when they are created.  
 Half moon shows specialization downwards, inheritance upwards.  
 The middle of a block contains the attributes, referenced as necessary via 'get/set' methods.  
 Attributes are read as "name : class".  
 Plural attribute names implies a set of objects. It does not imply how that set is implemented.  
 The bottom of a block contains parameterized methods.  
 This model is case sensitive. Class names start with caps, attributes with lower case.  
 Object creation methods ("new") are not implied or specified by this diagram, except for basic types.  
 Rectangle with list is enumerated type.





# Interaction Design System (IDS)

created.

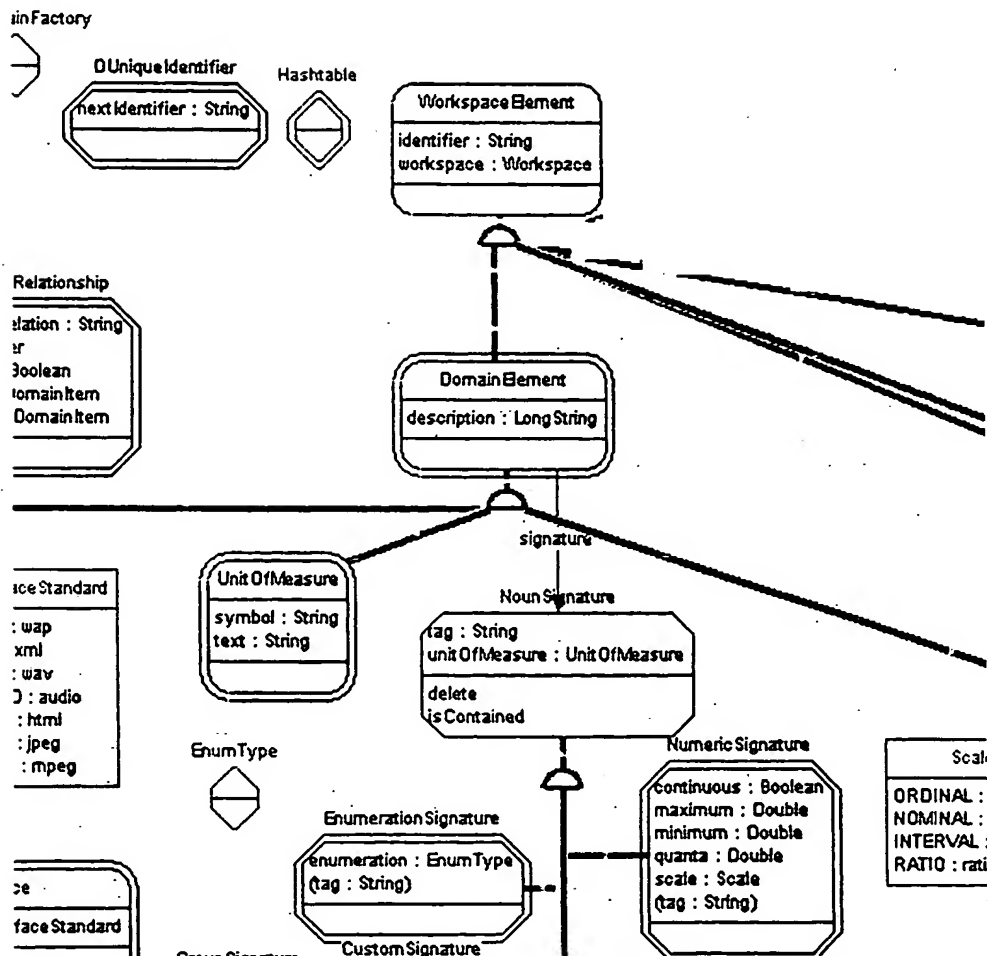
ods.

anted (list, array, etc.)

asic writeable attributes

architecture.net  
Model Type: Domain Independent Architec  
[ 2001SEP08 ]

OBJECTIVE: This is a class hierarchy whi  
forms the basis of a domain independent  
architecture for an Interaction Design Sys



nt Architecture

ierarchy which  
dependent  
Design System.

Utility Packages:  
util.Find  
util.URLFetch

These are the SRBICs  
InteractionCharacteristics

controlBandwidth : Long  
controlImportance : Long  
controlResolution : Long  
controlScope : Long  
displayBandwidth : Long  
displayImportance : Long  
displayObtrusiveness : Long  
displayResolution : Long  
displayScope : Long  
toBandwidth  
toResolution  
toScope

#### TaskTypes

NONPRIMITIVE : nonprimitive  
JUNCTION : junction  
PATH : path  
PRIMITIVE : primitive  
FINAL : final

#### InteractionR

availableInteractionDevi  
domainItem : DomainItem  
endTime : Calendar  
presentation : Presentation  
qualifyingPresentations  
startTime : Calendar  
task : Task

layoutQualifyingPresent  
possiblePresentationsTo

#### Scale

ORDINAL : ordinal  
NOMINAL : nominal  
INTERVAL : interval  
RATIO : ratio

#### Pres

device : InteractionDe  
interactionRequirements  
matchQuality : Interaction  
presentationElement :  
score : Long

PresentationE

ackages:  
ind  
JRLFetch

These are the SRBICs  
InteractionCharacteristics

controlBandwidth : Long  
controlImportance : Long  
controlResolution : Long  
controlScope : Long  
displayBandwidth : Long  
displayImportance : Long  
displayObtrusiveness : Long  
displayResolution : Long  
displayScope : Long

toBandwidth  
toResolution  
toScope

primitive

#### Interaction Requirement

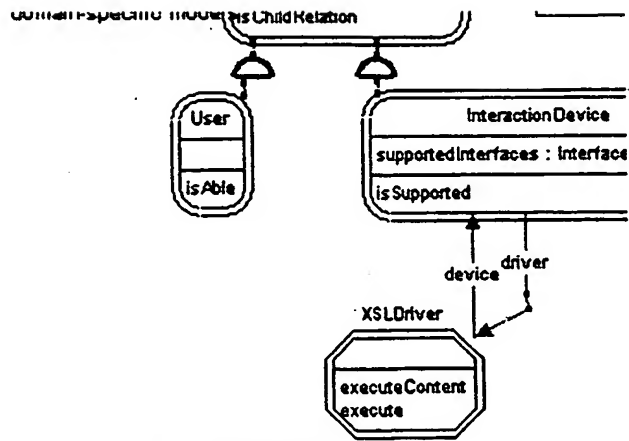
availableInteractionDevices : Interaction Device  
domainItem : DomainItem  
endTime : Calendar  
presentation : Presentation  
qualifyingPresentations : Presentation  
startTime : Calendar  
task : Task

layoutQualifyingPresentations  
possiblePresentationsToCSV

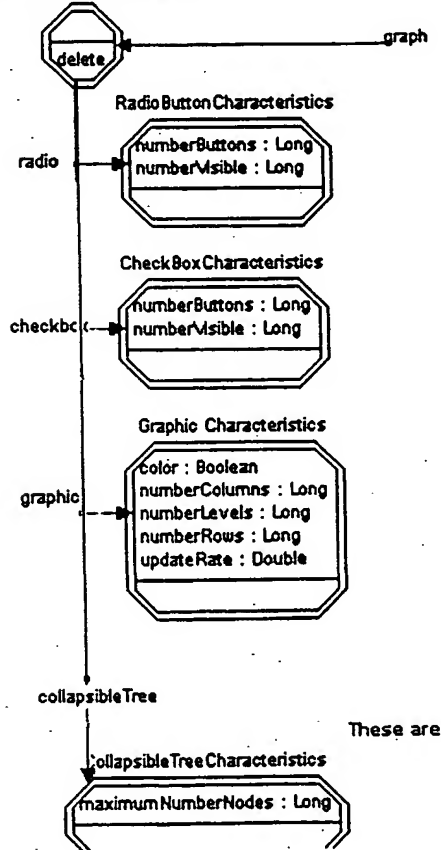
#### Presentation

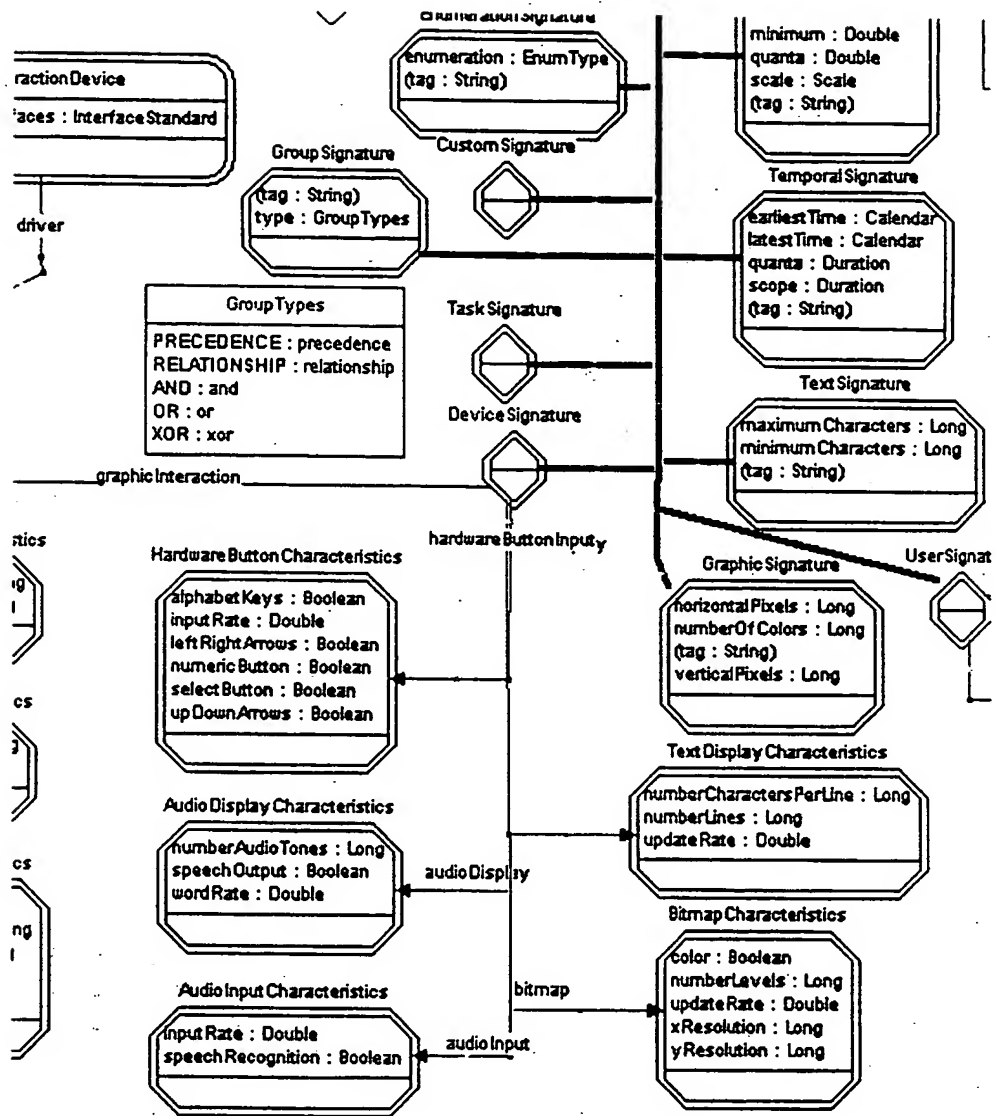
device : Interaction Device  
interactionRequirement : Interaction Requirement  
matchQuality : Interaction Characteristics  
presentationElement : Presentation Element  
score : Long

Presentation Element



#### Graphic Interaction Characteristics





These are the Device Characteristics v002

Boolean  
Double  
Double  
able  
s  
)

ORDINAL : ordinal  
NOMINAL : nominal  
INTERVAL : interval  
RATIO : ratio

Signature

: Calendar  
Calendar  
ation  
tion

Signature

characters : Long  
characters : Long

Relation

destination : Task  
relationship : Relationships  
source : Task

Relationships

RELATIONSHIP : relationship  
PRECEDENCE : precedence  
DESTINATION : destination  
SOURCE : source  
BOTH : both

Operations

AND : and  
OR : or  
XOR : xor  
IF : if  
LOOP : loop  
CASE : case

mat  
pre  
sco

required  
supporte  
supporte  
supporte  
  
getSpeci  
is Device  
is Presen  
is Primiti  
is Signati

presentation Content C

UserSignature

PhysicalAbilities Characteristics

physicalAbilities

CognitiveAbilities Characteristics

cognitiveAbilities

stics

: Long

TaskPrimitives

NONPRIMITIVE : nonprimitive  
RECEIVE : receive  
ASSERT : assert  
SELECT : select  
JUNCTION : junction  
INSTANTIATE : instantiate  
COMPARE : compare  
MONITOR : monitor  
CONTROL : control  
RETRACT : retract  
CHANGE : change  
DETECT : detect  
DIRECTATTENTION : directattention  
NAVIGATE : navigate  
ADJUST : adjust

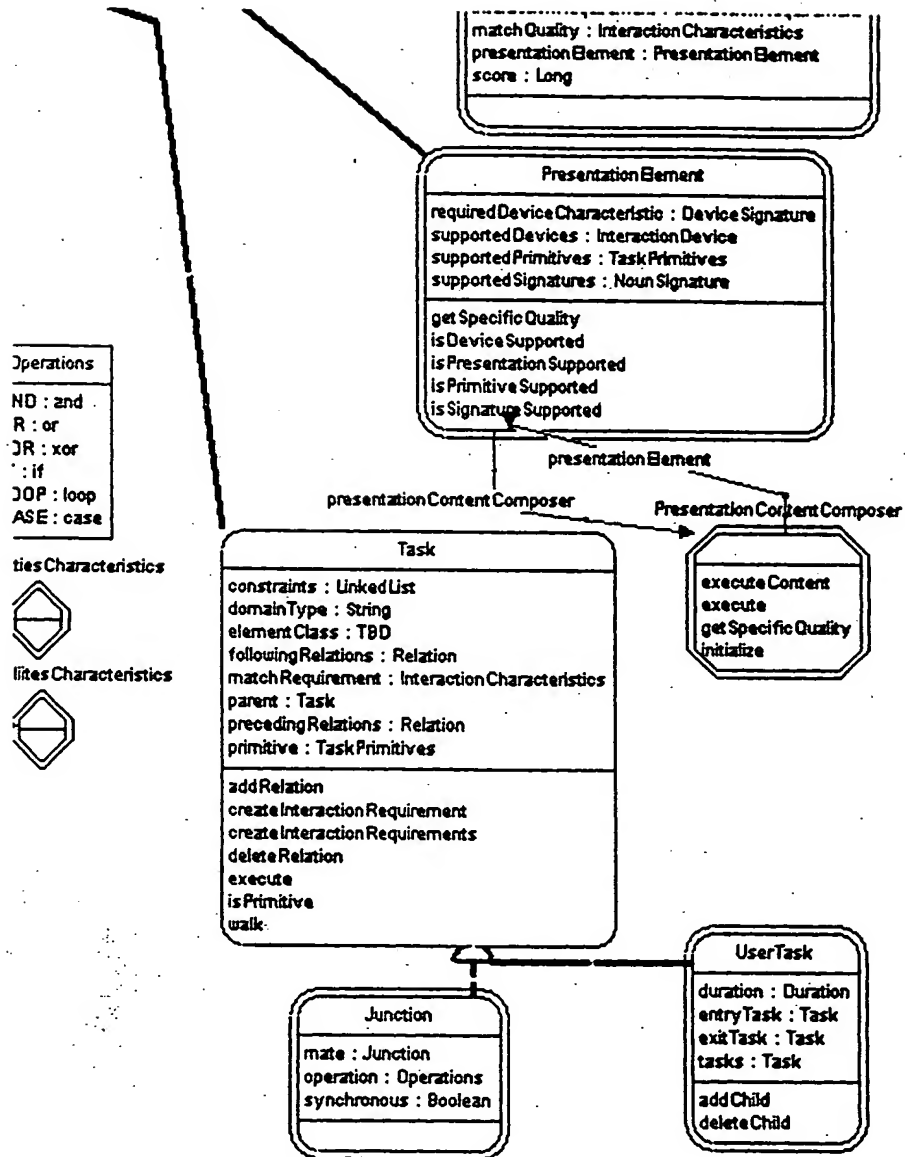
Task

constraints : Linked List  
domainType : String  
elementClass : TBD  
followingRelations : Relation  
matchRequirement : InteractionCl  
parent : Task  
precedingRelations : Relation  
primitive : TaskPrimitives

addRelation  
createInteractionRequirement  
createInteractionRequirements  
deleteRelation  
execute  
isPrimitive  
walk

Junction

mate : Junction  
operation : Operations  
synchronous : Boolean



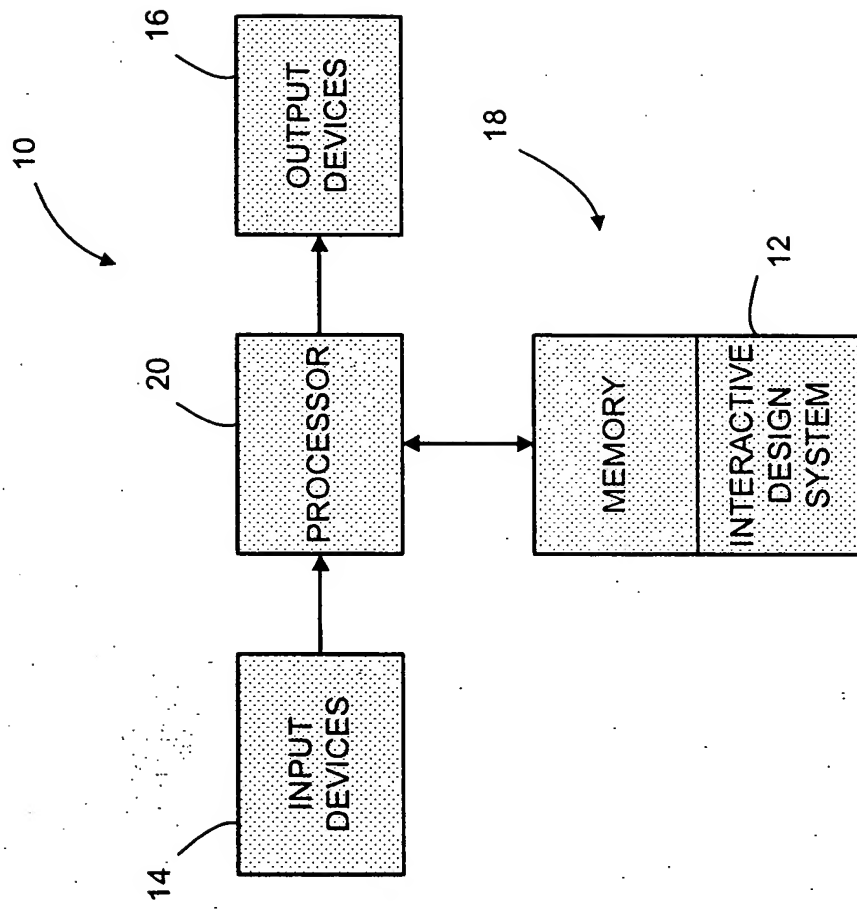


FIG. 1



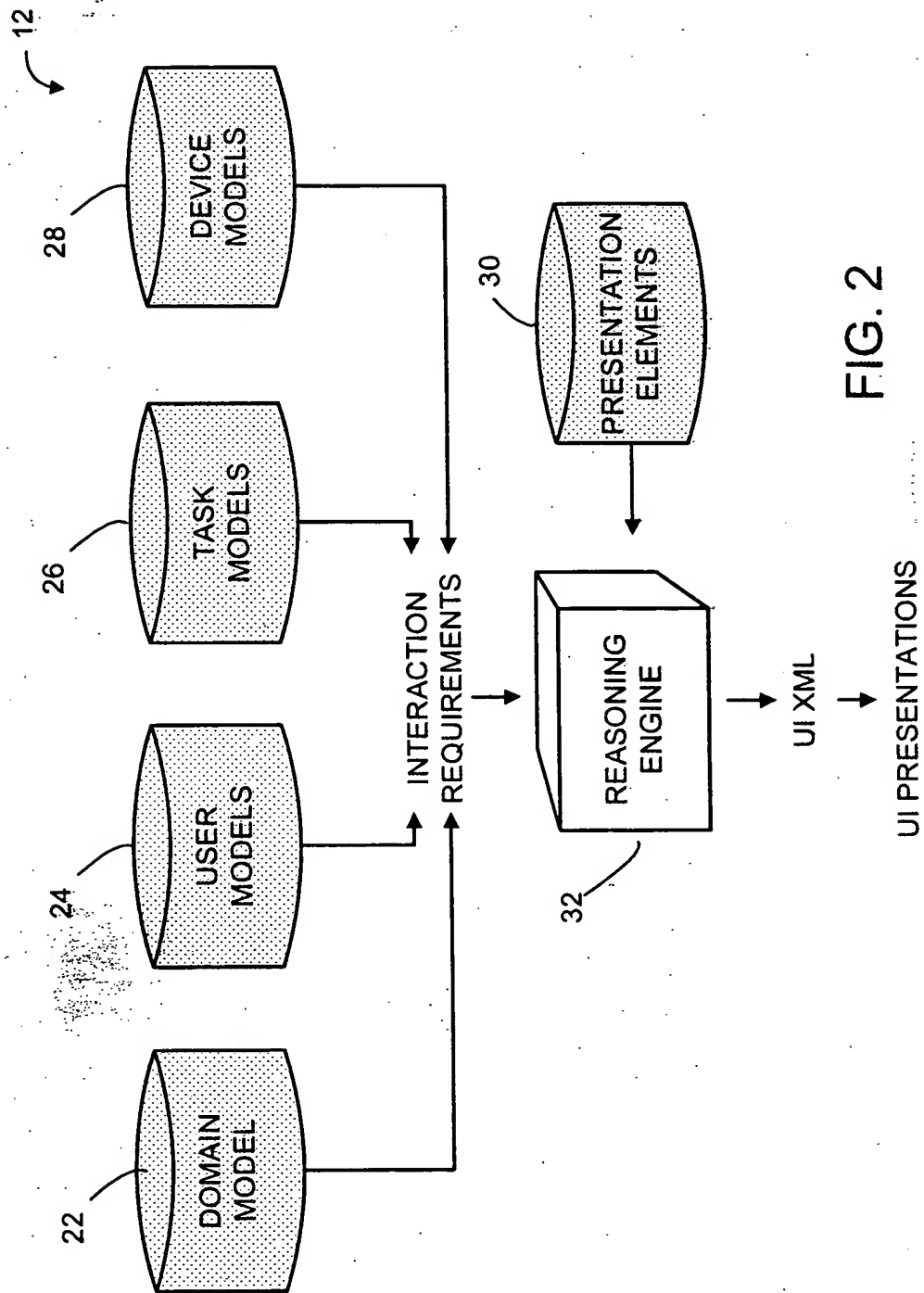


FIG. 2

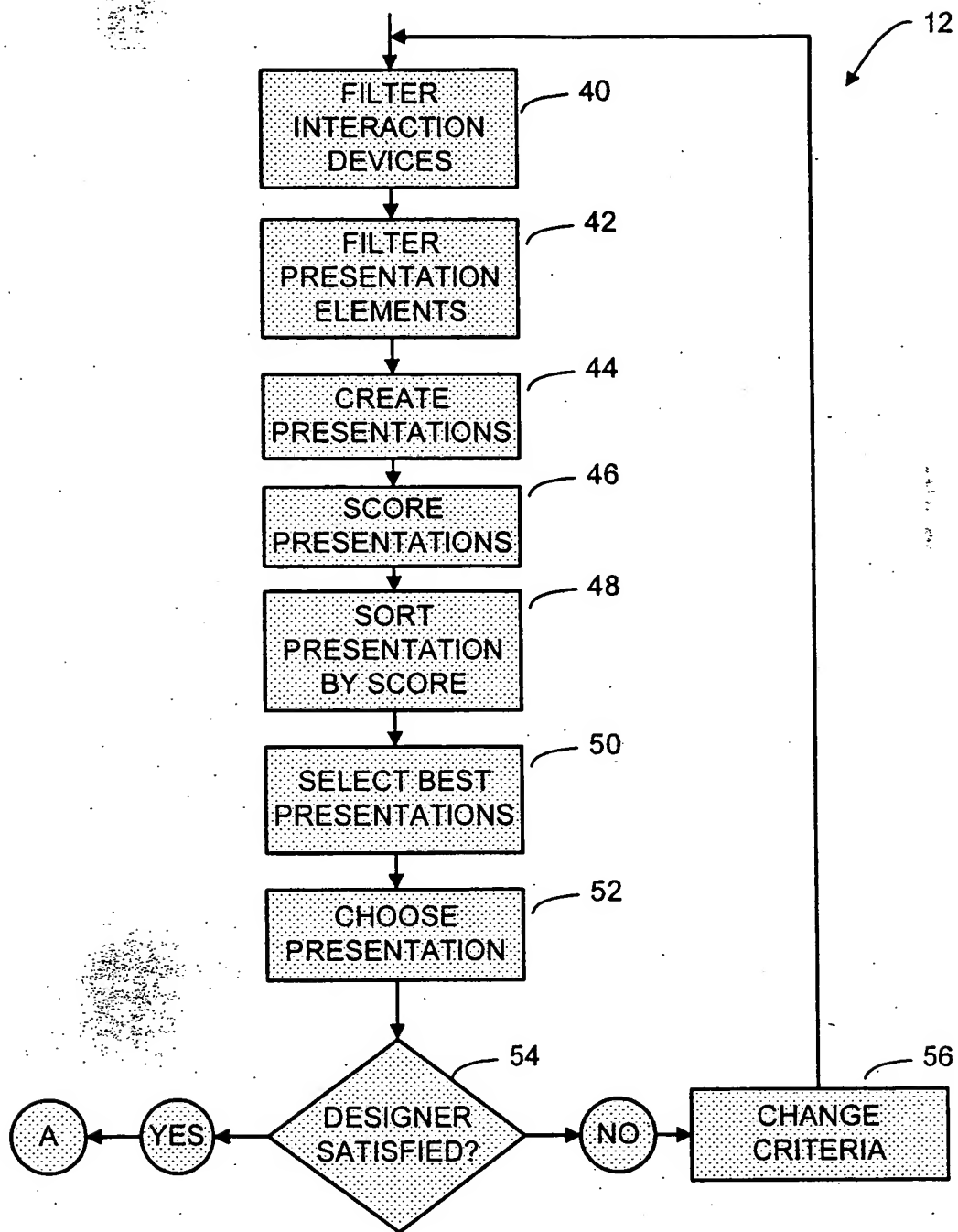


FIG. 3A

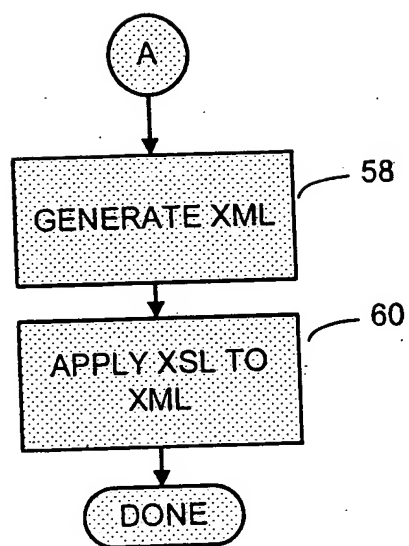


FIG. 3B